

CLAIMS

What is claimed is:

1. A method of compiling a program to be executed on a target
5 microprocessor with multiple execution units of a same type, the method comprising:
selecting one of the execution units for testing;
scheduling execution of diagnostic code on the selected execution unit;
and
10 scheduling execution of program code on remaining execution units of the same type
2. The method of claim 1, wherein the selection of the execution unit for testing utilizes an algorithm that assures testing of each of the multiple
15 execution units.
3. The method of claim 2, wherein the algorithm comprises a round-robin type algorithm.
- 20 4. The method of claim 1, further comprising:
setting a level of aggressiveness for scheduling the testing of the execution units.
- 25 5. The method of claim 4, further comprising:
applying an aggressiveness-dependent algorithm to determine when to schedule all available units for execution of the program code and when to schedule parallel execution of the program code and the diagnostic code.
- 30 6. The method of claim 5, wherein a lowest level of aggressiveness comprises turning off said testing.

7. The method of claim 1, wherein the multiple execution units of the same type comprise arithmetic logic units.
- 5 8. The method of claim 1, wherein the multiple execution units of the same type comprise floating point units.
9. The method of claim 1, wherein the multiple execution units comprise at least four execution units of the same type integrated onto the microprocessor integrated circuit.
- 10 10. The method of claim 1, wherein the scheduled diagnostic code performs diagnostic operations from a test pattern comprising operations with known expected results.
- 15 11. The method of claim 10, wherein the scheduled diagnostic code compares an actual result with a known expected result.
12. The method of claim 11, wherein the scheduled diagnostic code jumps to a fault handler if the compared results are different.
- 20 13. The method of claim 12, wherein the fault handler includes code to remove a faulty execution unit from use in executing code.
14. The method of claim 12, wherein the fault handler includes code to perform a system halt to prevent data corruption.
- 25 15. A computer-readable program product for execution on a target microprocessor having multiple execution units of a same type integrated thereon, the program product comprising:
30 diagnostic code configured to be executed on a selected execution unit of the multiple execution units; and
program code configured to be executed on remaining execution units of the same type.

16. The program product of claim 15, wherein the selected execution unit rotates between the multiple execution units such that each execution unit is tested.
- 5
17. The program product of claim 15, wherein the multiple execution units of the same type comprise arithmetic logic units.
18. The program product of claim 15, wherein the multiple execution units of the same type comprise floating point units.
- 10
19. The program product of claim 15, wherein the multiple execution units comprise at least four execution units of the same type integrated onto the microprocessor integrated circuit.
- 15
20. The program product of claim 15, wherein the scheduled diagnostic code performs diagnostic operations from a test pattern comprising operations with known expected results.
- 20
21. The program product of claim 20, wherein the diagnostic code compares an actual result with a known expected result.
22. The program product of claim 21, wherein the diagnostic code jumps to a fault handler if the compared results are different.
- 25
23. The program product of claim 22, wherein the fault handler includes code to remove a faulty execution unit from use in executing code.
24. The program product of claim 22, wherein the fault handler includes code to perform a system halt to prevent data corruption.
- 30

25. A computer-readable program product for execution on a target microprocessor having multiple execution units of a same type integrated thereon, the program product comprising:
diagnostic code configured to be executed on a selected execution unit of
5 the multiple execution units,
wherein the selected execution unit rotates between the multiple execution units such that each execution unit is tested, and
wherein said diagnostic code is further configured to be run in a background type process on a multi-threaded operating system.